Copy as Sent to Jon Steeby 11-20-95

1 sp~fsp. tg1

Originated by: <u>Thomas A. Genise</u>
Project No.: <u>R-706</u>
Date: <u>November 17, 1995</u>

Statement of Function:

This document describes the specific functional performance requirements for AutoSplit transmission system, which is based on the R747 transmission.

Approvals:	Signature	Date

J. Steeby
Engineering Supervisor - AutoSplit Systems Engineering

D. Gooch
Manager - Automation Product Engineering

S. Edelen
Manager - Automation

W. Batten
Manager - Automation Marketing

B. Vincent
Chief Engineer - Customer Quality

Exhibit 30

PROPRIETARY NOTICE

NOTICE TO PERSONS RECEIVING THIS DRAWING
AND/OR TECHNICAL INFORMATION

LINKED SPECIFICATION TREE

This diagram illustrates the hierarchy and linkage between specification which define the AutoSplit™ System.

```
                    ┌──────────────────────────┐
                    │       AutoSplit          │          Design Specifications
                    │ Product Design Specification │
                    └──────────────────────────┘
                                 │
          ┌──────────────────────┼──────────────────────┐
  ┌───────────────────┐  ┌───────────────────┐  ┌───────────────────┐
  │ AutoSplit Functional │  │ AutoSplit Electronic Control │  │ World-Wide Software │
  │ Performance Specification │  │   Unit Specification   │  │    Specification    │
  └───────────────────┘  └───────────────────┘  └───────────────────┘
                                 │
          ┌──────────────────────┼──────────────────────┐
  ┌───────────────────┐  ┌───────────────────┐          │
  │  Splitter Solenoid  │  │  Magnetic Speed   │          │          Procurement Specification
  │      PS-Spec        │  │      Sensor       │          │
  │                     │  │      PS-Spec      │          │
  └───────────────────┘  └───────────────────┘          │
                                              ┌───────────────────┐
                                              │  AutoSplit System   │
                                              │ Final Assembly Test │    Test Specifications
                                              │    Specification    │
                                              └───────────────────┘
```

REVISION LOG

DATE Section   Changes Made

TABLE OF CONTENTS

1   INTRODUCTION

1.1   Purpose

The intent of this performance specification is to document the functionality, diagnostics, and error handling methods of AutoSplit™ algorithms. *Italicized* words seen throughout this document are configuration performance parameters which are programmable in EEPROM. **Bold** print, other than section headers, indicate a status input into the system or variable names in the software.

1.2   Scope

This document describes the AutoSplit™ system level functional requirements. These requirements are defined using the intended electrical and mechanical constraints defined in specifications found in the Linked Specification Tree.

1.3   Acronyms and Abbreviations

ECU	Electronic Control Unit
EEPROM	Electronically Erasable Programmable Read Only Memory
CAN	Controller Area Network (also referenced as J1939)
ROM	Read Only Memory
RPM	Revolutions Per Minute
VBATT	Voltage, Battery (or Source)
VDC	Voltage, Direct Current
AutoSplit	The Eaton AutoSplit™ Transmission System
Super 10	Eaton Twin Countershaft, 10-speed transmission with range and splitter
R747	Next generation transmission family, of which the first model will be of the Super 10 configuration

2   LINKED SPECIFICATION LIST AND DOCUMENT CHANGE

2.1   Change Authority

All revisions or changes to this document will be maintained by the AutoSplit **System** Engineer. Revisions will be recorded in the revision log of this document.

2.2   Linked Specification List

- AutoSplit Product Design Specification
- Transmission Electronic Control Unit Specification
- Splitter Solenoid Valve Procurement Specification
- Magnetic Speed Sensor Specification
- AutoSplit System Final Assembly Test Specification

3   GENERAL DESIGN REQUIREMENTS

3.0.1   Brief Definition
The AutoSplit transmission system utilizes an R747 base transmission. All "splitter-only" shifts are fully automatic. In addition, all lever shifts feature automatic throttle manipulation and speed synchronization using J1939 engine control for easy lever shifting. This system results in clutch-less shifting with no throttle manipulation after a manual clutch start. A simple dash display informs the driver of the "best" gear, available gears, and which gear the system is synchronizing for when the transmission is in neutral.

3.0.2   AutoSplit Inputs/Outputs
The block diagram in Figure 3.0.2-1 below describes the required inputs and outputs (I/O) of the AutoSplit system. For a more complete understanding of the AutoSplit hardware and interfaces, please refer to the "AutoSplit Product Design Specification."
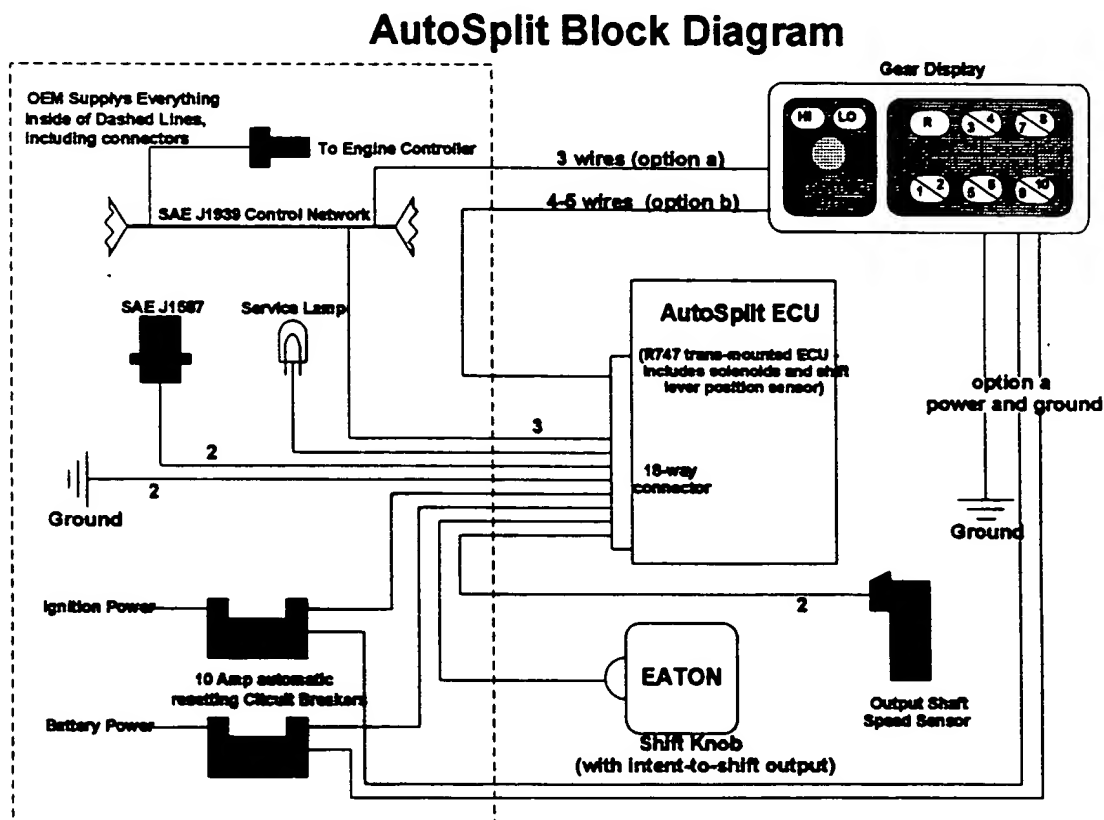
AutoSplit Block Diagram

Figure 3.0.2-1 - AutoSplit Block Diagram

3.0.3 AutoSplit Sequence of Events

Normal AutoSplit operation consists of both driver-initiated (lever) shifts and system-initiated, automatic (splitter) shifts. The normal shift sequence for each of these shifts is as follows:

Splitter-Only Shift

1) The system detects the optimal time to shift based on load, input_speed, etc. At this instant, the system overrides cruise control, engine brakes, throttle, etc. via J1939 engine override commands, commands the splitter to neutral, and modulates the engine torque to allow splitter disengagement via J1939.

2) The system confirms splitter disengagement via a comparison of the ratio of input_speed to output_speed with the transmission ratio table. At this instant, the system implements anti-hunting routines, and modulates engine_speed via J1939 to synchronize the splitter for the target ratio.

3) When the system senses impending synchronous via input_speed and output_speed signals, it commands the splitter towards target gear engagement.

4) The system confirms splitter engagement via a comparison of the ratio of input_speed to output_speed with the transmission ratio table. At this instant, the system commands the engine to reapply torque via J1939.

5) When torque has been reapplied, the engine resumes control of the throttle, engine brakes, cruise control, etc.

Lever/Splitter Shift

1) The driver display flashes the available lever position to indicate it is "OK" to shift the lever to that position.

2) When a lever shift is desired, the driver pulls the lever to neutral while activating the intent-to-shift feature (TBD – either a momentary button, or a force detent in the knob or lever). At this instant, the system overrides the cruise control, engine brakes, throttle, etc. via J1939, commands the splitter to neutral, and modulates the engine torque to allow splitter and lever disengagement via J1939.

3) The system confirms transmission neutral via a comparison of the ratio of input_speed to output_speed with the transmission ratio table, and by the lever position sensor. At this instant, the system implements anti-hunt routines, commands the splitter to the engaged position for the new ratio, and modulates the engine_speed via J1939 to synchronize the transmission for the target gear ratio.

4) The driver moves the lever into the new position.

5) The system confirms the new gear engagement via a comparison of the ratio of input_speed to output_speed with the transmission ratio table. At this instant, the system

3

commands the engine to reapply torque via J1939.
6) When torque has been reapplied, the engine resumes control of
the throttle, engine brakes, cruise control, etc.

3.1 APPLICATION REQUIREMENTS AND CONSTRAINTS

3.1.1 Applications
AutoSplit includes all "On-Highway" RoadRanger-type applications.
Vocations include Pickup and Delivery, LTL, TL, large and small
fleets. Engine applications include all diesel engines approved
for On-Highway RoadRangers that provide for SAE J1939 CAN
communications.

3.1.2 Calibration Parameters
Reference Appendix A for calibration parameters.

3.2 PERFORMANCE REQUIREMENTS

3.2.1 Vehicle Startup and Shutdown
The AutoSplit system will become active (power-up) whenever
v_ignition is asserted. Once **v_ignition** is asserted, the AutoSplit
system will not be disabled (power-down) until **v_ignition** is de-
asserted and **output_speed** < *min_output_spd.* No calibration or data
storage is required.

Upon **v_ignition** asserted, and **output_speed** < *min_output_spd,* the
system will command the splitter to the position indicated by the
splitter start gear selector button/switch on the driver display.
Also upon power-up, all of the driver display lights including the
service lamp will light for a one second (TBD) period to facilitate
a "lamp check."

Upon **v_ignition** de-asserted and **output_speed** > *min_output_spd,* the
system will remain in the current gear. When **output_speed** <
min_output_spd, the system will then power down.

3.2.2 AutoSplit Modes
Only one mode of operation is available with the AutoSplit system
and is available whenever the system is active, or powered-up.
This mode is explained in sections 3.2.2.1 to 5 below.

3.2.2.1 Driver Operating Procedure
To start from rest, the driver depresses the manual conventional
clutch and engages the desired lever position, as in a manual
transmission. If the selected splitter starting ratio is low
(direct), the starting gears can be 1, 3, 5, etc. whereas if the
selected splitter starting ratio is high, the starting gears will

be one of 2, 4, 6, etc. The splitter starting is selected by a switch or push-button on the driver display. To start the vehicle, the driver releases the clutch as in a manual gearbox vehicle.

When the ECU determines that it is time for an upshift (or downshift), and the shift is a splitter-only shift (i.e., 1-2, 3-4, 4-3, etc.), the transmission automatically performs the shift.

However, a lever shift is different and is driver-initiated. For a lever upshift, the driver pulls the transmission lever to neutral while actuating the intent-to-shift switch. This switch may be "driver-active" like a momentary push-button, or "driver-passive" using a force detent switch in the knob or lever. Activating this switch causes the system to preselect the splitter to neutral and manipulate engine torque to achieve a driveline torque reversal via J1939 engine override control. Alternately, the driver can manipulate the throttle to reverse driveline torque and pull the lever to neutral. Upon sensing neutral, the system commands the splitter to the target ratio engagement position and commands the engine (via J1939) to go to the speed for synchronous for the next gear. The driver engages the lever into the next gear position. The system detects engagement and ramps the engine torque back up to what the driver is demanding.

Note that the vehicle master clutch should not be held depressed during lever shifting as this would interfere with the engine control of synchronous. However, if the driver does depress the clutch at any time (clutch state reported by engine via J1939), the engine control would be returned to Engine Follower mode (no override) until the clutch is again engaged. The driver could momentarily depress the clutch to move the lever to neutral without affecting the engine speed control. As soon as the system recognizes that the clutch is reengaged, Engine Speed Control can resume. If the driver "double-clutched" during a lever shift, the transmission would behave in the same manner as a manual RoadRanger transmission.

3.2.2.2 Driver Display Function
The driver display as pictured in Figure 3.0.1-1 has 10 indicators that can individually illuminate each of the 10 forward gear numbers in their respective shift pattern position. When the vehicle is moving and the transmission is engaged in a gear, the indicator corresponding to that gear is steadily lit (not flashing). Also when in a gear and moving, the display flashes (blinks) an indicator other than that one already lit if the system determines a lever shift to that position is possible and allowable. Note that for a splitter-only shift the target gear would not need to flash since the system would automatically initiate that shift. When the driver initiates a lever shift and

brings the shift lever to neutral, and the neutral sensing routine confirms neutral, the gear indicator that was reporting the engaged gear turns off and the flashing indicator that was reporting an allowable shift continues flashing for that target gear, indicating that the system is now directing the engine to go to the speed to create a synchronous condition for the new gear. If the vehicle speed changes sufficiently while the transmission is in neutral for the system to change its selection for the "best" gear, the indicator corresponding to the new gear will begin flashing and the system will command the engine to synchronous for the new gear. When the driver engages the lever in a new position, and the system senses engagement of that new gear, and the indicator corresponding to that new position will be illuminated steadily.

The driver display also includes a push-button switch to select which splitter gear the system will engage at rest. Besides allowing the driver to start in either splitter ratio, this gives the system two reverse ratios. Two indicators on the display inform the driver whether "HI" or "LO" splitter starting ratio has been selected.

3.2.2.3 Shift Scheduling

AutoSplit is unique from other automated mechanical transmissions in that its electronically-enhanced shift operations consist of both driver-initiated (lever) shifts and system-initiated, automatic (splitter) shifts. Therefore, a shift scheduling routine is used that recognizes the different types of shifts and takes different action for each. Figure 3.2.2.3-1 attempts to summarize this shift schedule. Note that AutoSplit does not permit driver-chosen skip shifting. However, the system may select a skip shift under certain circumstances (i.e., a lever downshift is initiated and the driver allows the vehicle to slow enough to choose the next lowest splitter position while in neutral).

In Figure 3.2.2.3-1, the automatic splitter-only shift points are throttle-modulated. That is, the greater the throttle opening, the higher the shift point in speed. In Figure 3.2.2.3-1, these shifts are the 1-2, 2-1, 3-4, 4-3 shifts, and so on. The allowable shift speeds for each gear are noted by the arrows. For example, for the 1-2 shift, the arrows pointing down at the upper end of the first gear "speed line" from 1500 rpm to 1800 rpm signify that 1-2 upshifts are initiated from 1500 to 1800 rpm input speed depending upon throttle (see Figure 3.2.2.3-2). Note that the arrows point down to indicate that the input speed must come down to complete the 1-2 shift. See the equations later in this section to determine shift points and shift types.
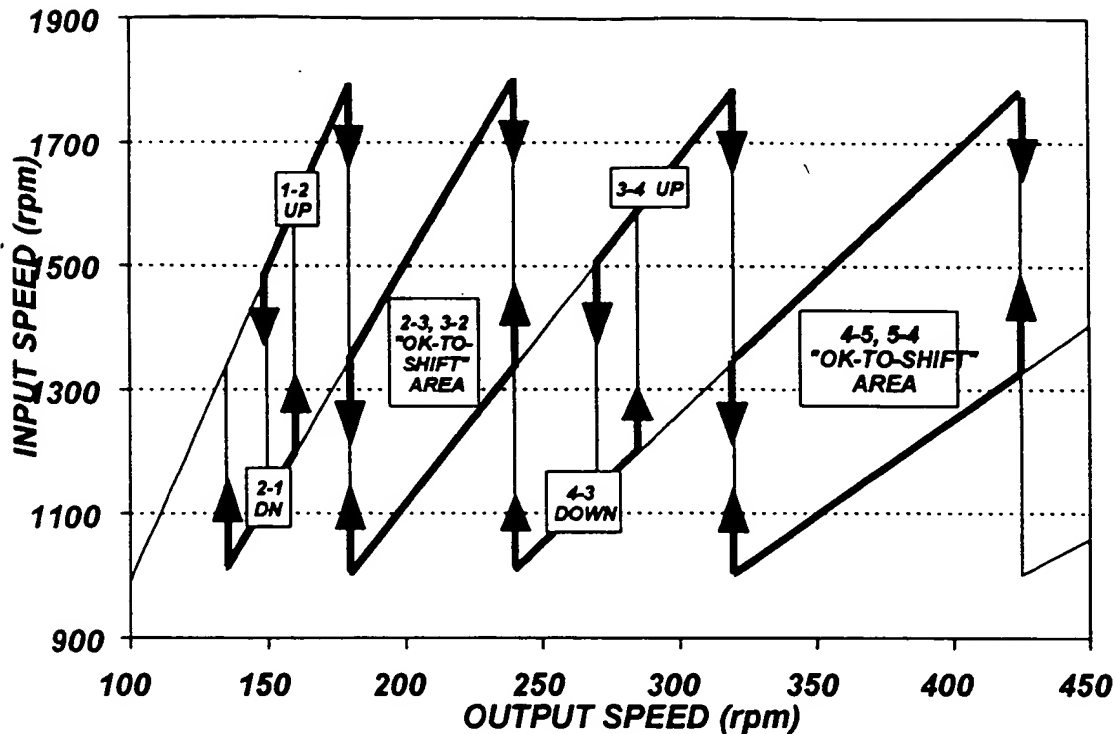
Figure 3.2.2.3-1 - AutoSplit Shift Schedule

The lever shifts on Figure 3.2.2.3-1 indicate different information than the splitter shifts do. The lever shifts are the 2-3, 3-2, 4-5, 5-4 shifts, and so on. Note that these are called "ok-to-shift areas." This means that these areas indicate allowable speeds for those lever shifts by the system. It is up to the driver to determine the input speed at which those shifts are actually initiated. For example, if the transmission is in 2nd gear (4th gear, 6th gear, etc.), the input speed is above 1375 rpm, and neutral is sensed, the system recognizes it as a 2-3 lever shift. Once neutral is seen, the splitter is shifted to the low position and the engine is commanded to a speed that achieves synchronous for the new gear, which is 3rd. The calibrations on this figure are for an application with an 1800 rpm governed speed engine.

Figure 3.2.2.3-2 illustrates this shift logic in a different manner. Note the "notch" in the splitter upshift line above 90 percent driver demand (percent throttle). This is the software simulating a "ride-through-detent" feature at wide-open-throttle.

7

Each of the "curves" in Figure 3.2.2.3-2 only applies when the transmission is in the appropriate gear for that line. For example, the "lever upshifts" line only applies when the transmission is in gear 2, 4, 6, or 8. When the transmission is operating in one of these gears and the input speed is above 1375 rpm, the light on the dash display corresponding to the next higher lever position flashes indicating to the driver that it is "OK-to-shift" to the next highest gear. If the driver moves the lever to neutral (in gears 2, 4, 6, or 8) and the input speed is not above 1375, the system will synchronize (command engine to synchronous speed) for the gear that was just disengaged. Note the system has no indication of driver intentions other than **input_speed** and **output_speed**.
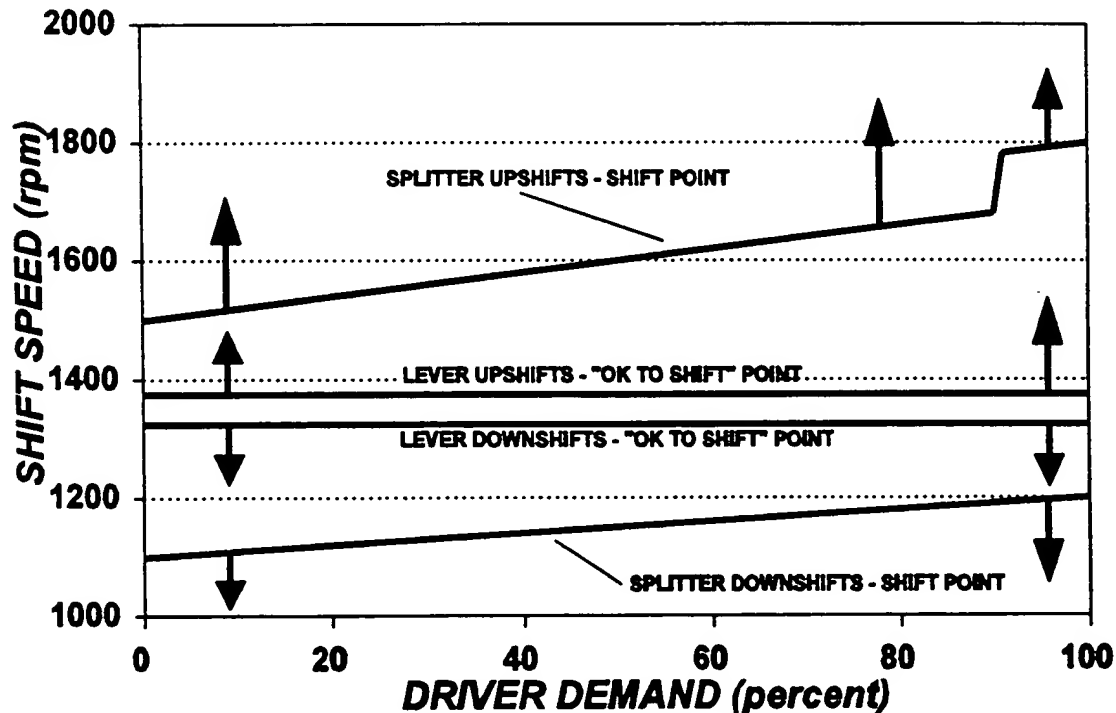

Figure 3.2.2.3-2 - AutoSplit Shift Calibrations Sample

"Driver demand" (**pct_demand_at_cur_sp**) used in Figure 3.2.2.3-2 is a variable that is reported by the engine via J1939. Under most conditions, it is identical to percent throttle. The exception is

the condition where the vehicle cruise control is engaged and the throttle pedal is released.   In this case, it is the amount of power that the cruise control is demanding to maintain its speed setting.   This provides modulated shift points for splitter shifting while in cruise control. AutoSplit is fully compatible with vehicle cruise control.   Upshifts and downshifts can be executed (both splitter and lever) without disengaging or having to resume cruise control after the shift.

The **upshift_point** and **downshift_point** for splitter-only shifts are determined by **pct_demand_at_cur_sp**, which is a variable reported by the engine via J1939 representative of throttle position, as discussed above, and is determined by the following algorithm:

> **auto_up_rpm** = *auto_up_lo_base* +
> 　　　(*aut_up_rpm* – *auto_up_lo_base*) x **pct_demand_at_cur_sp**

> If (**pct_demand_at_cur_sp**) > 90
> 　　　**auto_up_rpm** = **auto_up_rpm** + auto_rtd_offset

> **auto_dn_rpm** = *auto_dn_lo_base* +
> 　　　(*aut_dwn_rpm* – *auto_up_lo_base*) x **pct_demand_at_cur_sp**

Then a check is made to determine if one of the two points above should become a manual shift point. Specifically:

> If **last_known_gear** = 1, 3, 5, 7, or 9, then,
> 　　　**auto_dn_rpm** = *manual_dn_base*

> else,
> 　　　**auto_up_rpm** = *manual_up_base*

> **upshift_point** = **auto_up_rpm**

> **downshift_point** = **auto_dn_rpm**

Note that the same shift point variable names are used for both splitter-only shifts and lever shifts. The routine that then selects the new gears based on these points, then recognizes what type of shift, or **shift_init_type** - AUTO or MANUAL - the point represents. A routine is used to determine this type based on the proposed **new_gear** and the currently engaged gear (**initial_gear**). Specifically,

```
If   (the new_gear =  1   and the initial_gear =  2, or
      the new_gear =  3   and the initial_gear =  4, or
      the new_gear =  5   and the initial_gear =  6, or
      the new_gear =  7   and the initial_gear =  8, or
      the new_gear =  9   and the initial_gear = 10, or
      the new_gear = 10   and the initial_gear =  9, or
      the new_gear =  8   and the initial_gear =  7, or
      the new_gear =  6   and the initial_gear =  5, or
      the new_gear =  4   and the initial_gear =  3, or
      the new_gear =  2   and the initial_gear =  1),
```

then,
  shift_init_type = AUTO

else,
  shift_init_type = MANUAL

Anti-hunt strategies in place after a recent upshift may change the value of **downshift_point**. Refer to section 3.2.2.5 for anti-hunt strategies.

3.2.2.4 Shiftability Algorithm
"Shiftability" is an algorithm that calculates the feasibility and/or desirability of a given shift and takes action based on those calculations. Two "levels" of shiftability will be used in the AutoSplit transmission system. Both pertain specifically to splitter-only shifts. The first level is a "reactive" algorithm that will make changes once the shift has begun. The second level is "proactive" and will determine whether or not a shift is feasible or desirable before the shift is started.

3.2.2.4.1 Reactive Shiftability
The reactive shiftability algorithm monitors the output shaft acceleration (**output_speed_accel**) and modifies the gear engagement synchronous windows based on this acceleration. Specifically, if the vehicle deceleration (**output_speed_accel**) becomes more negative than a certain value, the synchronous window for that gear will be widened in proportion to that deceleration. This will allow earlier, farther out-of-synchronous engagement. This algorithm is only used for splitter-only upshifts since the splitter is engaged

immediately after neutral is sensed during lever shifts.

Output_speed_accel must be heavily filtered to be used for this algorithm. See the variable **dos_filtered** in the next section for the filter method and values. **Dos** stands for derivative of output_speed. **Dos_filtered** is then multiplied by the gear ratio of the target gear to obtain the variable **dgos**, which is an acronym for derivative of gear_ratio times output_speed. Then **dgos** is used in the following algorithm to determine the upshift synchronous "window". The upper and lower speed windows (the synchronous error corrected to the input shaft) are in a table with different windows for each ratio. These windows represent the speed differential from synchronous at which the splitter may be engaged (i.e., + or - 60 rpm). The reactive shiftability algorithm only affects the upper window value because it is the speed value that allows engagement of the splitter for splitter-only upshifts. The algorithm is as follows:

sp_offset_pos = -((**dgos** x *sp_slope*) + *sp_zero_point*)

If (**sp_offset_pos** < *splitter_sync_table*[**last_known_gear**]),
then **sp_offset_pos** = *splitter_sync_table*[**last_known_gear**].

If (**sp_offset_pos** > *spltr_offset_max*]),
then **sp_offset_pos** = *spltr_offset_max*].

sp_offset_neg = *splitter_sync_table*[**last_known_gear**]

Then, to determine if the speeds are within the synchronous windows, the following logic is used:

gos_spltr = *gear_ratio*[**destination_gear**] x **output_speed**

input_speed_modified =
        input_speed + (**input_speed_accel**/1000/*splitter_tc*)

If ((input_speed_modified < (gos_spltr + sp_offset_pos)) and,
  ((input_speed_modified > (gos_spltr - sp_offset_neg)),
then **splitter_within_sync** = TRUE.

3.2.2.4.2 Proactive Shiftability
The proactive shiftability algorithm will follow the following logic rules. Note that it takes into account the reactive shiftability algorithm, as well as the presence of engine retarding devices present (i.e., an engine compression brake, engine inertia brake, etc.). The algorithm general rules are listed below and then the actual calculation method is included later in this section.

1) <u>Vehicle Deceleration Limitation Check</u>
   a) The algorithm determines if the vehicle will slow down too much during the upshift to make the shift.
      If it will, it:
   b) determines if the vehicle can make the upshift with the help of the engine brake, and/or engine inertia brake - if the vehicle/transmission is so equipped.
      If it cannot, it:
   c) inhibits the upshift until conditions change enough to make the shift feasible.

2) <u>Vehicle Torque Limitation Check</u>
   a) Next, the algorithm determines if the vehicle has enough torque available in the proposed next gear to sufficiently accelerate.
      If it does not, it:
   b) inhibits the upshift until conditions change enough to make the shift desirable and prevent lugging or hunting.

3) <u>Vehicle Weight Calculation</u>
   This is a variable necessary for calculations 1) and 2) above. It is also useful for many other applications. Vehicle weight (GCW) will be used in this transmission in an algorithm that will modify shift calibrations as a function of GCW to have uniformly smooth shifts across the spectrum of possible GCW values. This is explained elsewhere in this document.

The proactive shiftability algorithm is taken from the AutoShift software and modified for lever-shifted automated transmissions (a modification not yet developed as of August 1995). The algorithm is further explained below.

Shiftability theory relies upon two dynamic pieces of data for input: engine torque (**actual_engine_pct_trq**, as reported from the J1939 data link), and vehicle acceleration (represented as **output_speed_accel**, derived from **output_speed**). The torque signal from the engine can vary quite quickly (every 20-40 milliseconds). The **output_speed** signal is greatly affected by torsional vibrations in the driveline. When **output_speed_accel** is calculated from the **output_speed** signal, the effect of the torsional vibrations is increased an order of magnitude.

To make the **output_speed_accel** signal usable, it must be filtered in two stages because of the high amount of filtering required and microprocessor accuracy limitations. Then, the engine torque (**actual_engine_pct_trq**) signal is filtered by an amount that keeps

it in phase with the heavily filtered **output_speed_accel** signal. Note that when a signal is mathematically filtered, the filtered signal lags real time by some amount. If that filtered signal is used in calculations with another signal, the second signal should be filtered by an amount that creates the same time lag, or phase lag. The engine torque signal (**actual_engine_pct_trq**) is reported by the engine via J1939 and is already filtered to some degree. The amount that the engine torque signal is further filtered is derived empirically. Again, it is not critical how much the signals used for shiftability are filtered as long as they are in phase with each other and adequately filter the effects of torsional vibrations.

These two filtered signals - **actual_engine_pct_trq** and **output_speed_accel** - are then used to calculate the various parameters in shiftability. The calculated results also need to be filtered further to remove the effects of torsionals and other signal variations.

The AutoSplit transmission must interrupt driveline torque to make a shift. In high road load conditions (i.e., grades, high rolling resistance, etc.), the vehicle may decelerate too quickly to successfully complete upshifts in the lower gears. Therefore, the ability to predict this vehicle deceleration at zero torque is the key to overcoming this issue. Figure 3.2.2.4.2-1 illustrates the situation in terms of speed signals.

The equation used to predict <u>vehicle deceleration at zero torque</u> is shown below. At this point, it is assumed that the gross combined weight, "W", is known by the algorithm. The procedure to determine "W" will be discussed later in this section.

$$\text{For } T_2 = 0: \quad A_2 = A_1 - \frac{T_1}{C \cdot W}$$

Where:
A_1 = vehicle acceleration at time I (ft/sec^2)
T_1 = torque at wheel at time I (lb-ft)
C = wheel rolling radius (ft)/grav. const. (ft/sec^2)
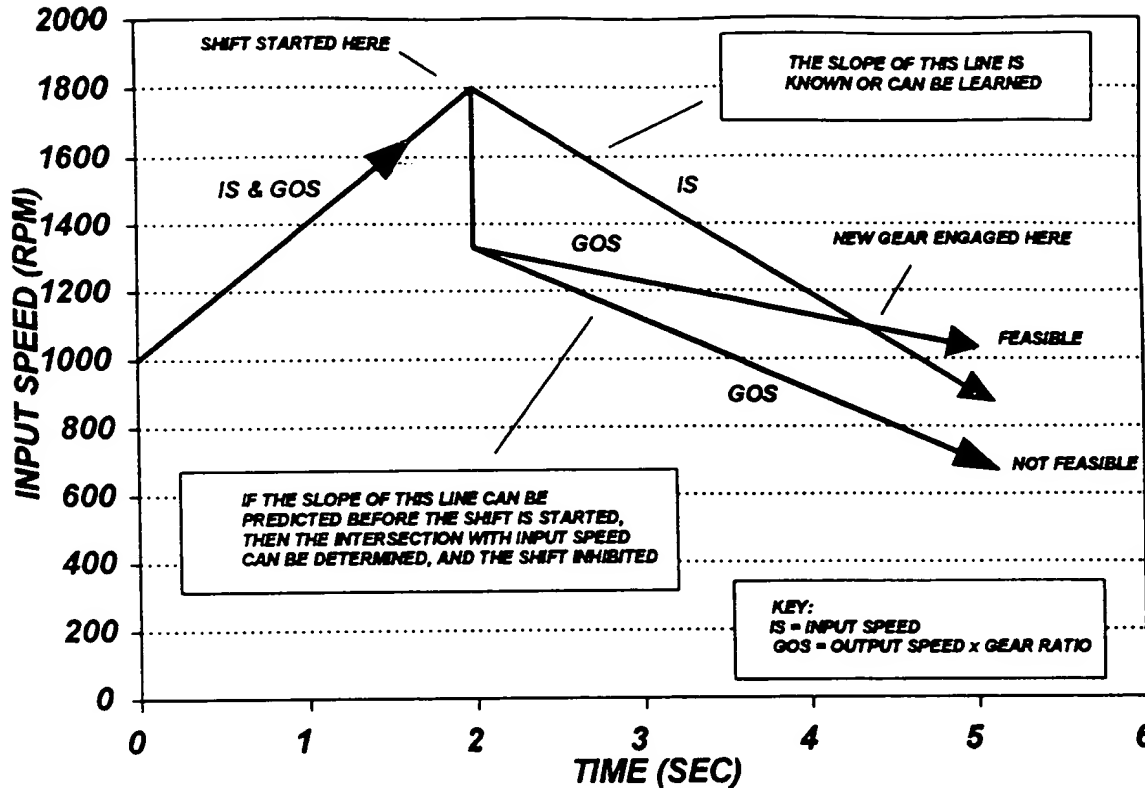W = gross combined weight (lb)

Figure 3.2.2.4.2-1 - Predicting Vehicle Deceleration

First, the variables that must be calculated and filtered every 10 milliseconds are derived.

a)   From **output_speed** (rpm), calculate **output_speed_accel** (rpm/sec) and filter it at a .97 filter level. This filter takes 97% of the current value and 3% of the new raw value. The result of this first filter is **lpf_output_accel** (rpm/sec).

   lpf_output_accel (rpm/sec) =
              (.97 · **lpf_output_accel**) + (.03 · **output_speed_accel**)

b)   Filter **lpf_output_accel** again at a .97 filter level and get **dos_filtered** (rpm/sec).

   dos_filtered (rpm/sec) =
              (.97 · **dos_filtered**) + (.03 · **lpf_output_accel**)

14

Note that "**dos**" is an acronym for derivative of output speed. The above filter values were determined empirically to result in a stable usable signal. The software designer must take care to maintain the proper amount of precision in these filter calculations, since the amount of filtering is heavy. Development of the algorithm found that high precision was especially needed in the second filter (**dos_filtered**) to prevent loss of precision.

c) Filter **net_engine_pct_trq** at a .98 filter level. This filter level was determined empirically to keep the engine torque value in phase with the heavily filtered **dos_filtered** parameter.

   net_engine_pct_trq =
            actual_engine_pct_trq - nominal_friction_pct_trq

The variables **actual_engine_pct_trq** and **nominal_friction-_pct_trq** are reported by the engine via J1939.

   eng_percent_torque_filtered =
                     (.98 · **eng_percent_torque_filtered**)
                        + (.02 · **net_engine_pct_trq**)

d) The acceleration of the input shaft can be calculated from **dos_filtered**. This is done instead of calculating input shaft acceleration directly from **input_speed** and heavily filtering it. It guarantees that the signals stay in phase and reduces code. Note that **input_speed_accel_calc** can be calculated from **dos_filtered** since the input shaft acceleration is only of interest while the transmission is in gear for these shiftability calculations (since the determination of shift feasibility and/or desirability is made <u>before</u> the shift is initiated).

   input_speed_accel_calc (rpm/sec) =
                           dos_filtered · *gear_ratio*

The remainder of the shiftability calculations can be done at a lower frequency (i.e., every 40 msec).

At this point, the values needed to predict vehicle acceleration, A_1 and T_i, are calculated. First, torque at the wheels must be calculated by working from the engine back to the wheels.

e) The torque used to accelerate the engine is calculated for later subtraction from engine torque, since it is not usable vehicle torque.

> torq_to_accel_eng (lb-feet) =
>   input_speed_accel_calc (rpm/sec) · *k6_ability*

where,
  k6_ability =
      [(2 · π) (rad/rev) / 60 (sec/min)] · I (lb-ft-sec²)

  I = rotating inertia of engine
    = 3.1 lb-ft-sec² Detroit Diesel Series 60 engine

(Note: units of *k6_ability* are rad-min-lb-ft-sec/rev)

The above equation is essentially $T = I \cdot \alpha$. Note that this inertia includes the engine, flywheel, clutch, and input of the transmission. This number is derived empirically by monitoring the J1939 torque reported value during wide open throttle engine acceleration with the transmission in neutral, measuring the resulting input shaft acceleration, and calculating "I.")

f) Then, engine torque available at the flywheel is calculated as follows:

> engine_torque (lb-ft) =
>   [eng_percent_torque_filtered (%) ·
>   engine_config.reference_trq (lb-ft/100)]
>   - [torque_accessories (lb-ft)
>   + torq_to_accel_eng (lb-ft)]

where **engine_config.reference_trq** is a reference torque value reported by the engine upon power up from which all other engine-reported torque values are referenced. The value, **torque_accessories**, is a variable that can change and is determined by another portion of this algorithm to be discussed later in this report.

16

g) The vehicle acceleration before the shift (**bs**), A_1, is now calculated from **dos_predicted** using a constant, *k1_ability*, that is a function of axle ratio and wheel rolling radius.

A_1 = **vehicle_accel_bs** (ft/sec^2)
     = **dos_filtered** (rpm/sec) · *k1_ability*

where,
  k1_ability = [2π (rad/rev) / 60 (sec/min)] ·
                 [rolling radius (ft) / axle ratio]

(Note: units of *k1_ability* are (rad-min-ft)/(rev-sec))

h) The torque at the wheels before the shift, T_1, which corresponds to A_1 is calculated.

  torque_at_wheels (lb-ft) =
                engine_torque · *gear_ratio* · *axle_ratio_cal*

where,
  axle_ratio_cal = axle ratio · overall driveline efficiency

(overall driveline efficiency = .91 in AutoShift software)

I) The vehicle acceleration at zero torque, A_2, (**after shift, or as**) is now calculated.

A_2 = **vehicle_accel_as** (ft/sec^2)
     = **vehicle_accel_bs** (ft/sec^2)
        - [**torque_at_wheels** (lb-ft) / **gcw_cal** (lb-sec^2)]

where,
  gcw_cal (lb-sec^2) =
        [W(lb) · roll. rad (ft)] / grav. const. (ft/sec^2)

  vehicle_accel_bs = A_1 = vehicle acceleration before the
                                shift begins

(gravitational constant = 32.17 (ft/sec^2)

(Note that "W" is the **gross_combined_weight** (GCW) determined by this algorithm as explained later in this report.)

j) The **vehicle_accel_as** is then converted back to rotational
    acceleration of the output shaft, to get **dos_predicted_raw**.
    Again, "**dos**" is an acronym for derivative of output speed.
    Therefore **dos_predicted_raw** is the predicted output shaft
    acceleration at zero torque (during a shift), and "raw" means
    that it is an unfiltered variable. (Note that the letters
    "**as**" for after shift could have been more accurately "**ds**" for
    during shift. However, this specification will use "as" to
    be consistent with AutoShift software variable names).

> **dos_predicted_raw** (rpm/sec) =
> $$\textbf{vehicle\_accel\_as} \ (\text{ft/sec}^2) \ / \ k1\_ability$$

where,

> $k1\_ability$ (rad-min-ft/rev-sec) is the same as in step (g)
> on the previous page.

(Note: The variable above is called **dos_predicted_raw** because
it is a single "raw" value of **dos_predicted** calculated during
one call to the software module. This variable will be
filtered next.)

k) **Dos_predicted_raw** is filtered to minimize the effects of
    torsional vibrations.

> **dos_predicted** (rpm/sec) =
> $$(.96 \cdot \textbf{dos\_predicted}) + (.04 \cdot \textbf{dos\_predicted\_raw})$$

Again, a high precision filter routine is needed.

Dos_predicted_raw is derived from transmission output shaft
acceleration and the engine torque value. Both of these values are
filtered before the calculation. Then, **dos_predicted_raw** is
filtered even more, creating **dos_predicted**. This heavy filtering
is needed because the two signals used are both affected by
driveline torsionals, and therefore this effect gets "amplified" in
the result, **dos_predicted_raw**, which must be further filtered.
Filtering creates a time lag from real time. Fortunately,
dos_predicted is a value that reflects the current vehicle road
loads, which do not change appreciably over a few seconds. Note
that **dos_predicted** is most valuable in the lower gear ratios, where
torsional vibrations are greatest in magnitude.

Now that **dos_predicted** is calculated, what is its purpose?

Again, **dos_predicted** is the predicted rotational acceleration of
the transmission output shaft during zero driveline torque (a

shift) in units of rpm/sec.

The software must calculate a "limit" (or limits) for which if
dos_predicted is more negative (faster deceleration), the shift is
not feasible.   The   problem   is   illustrated   graphically   in
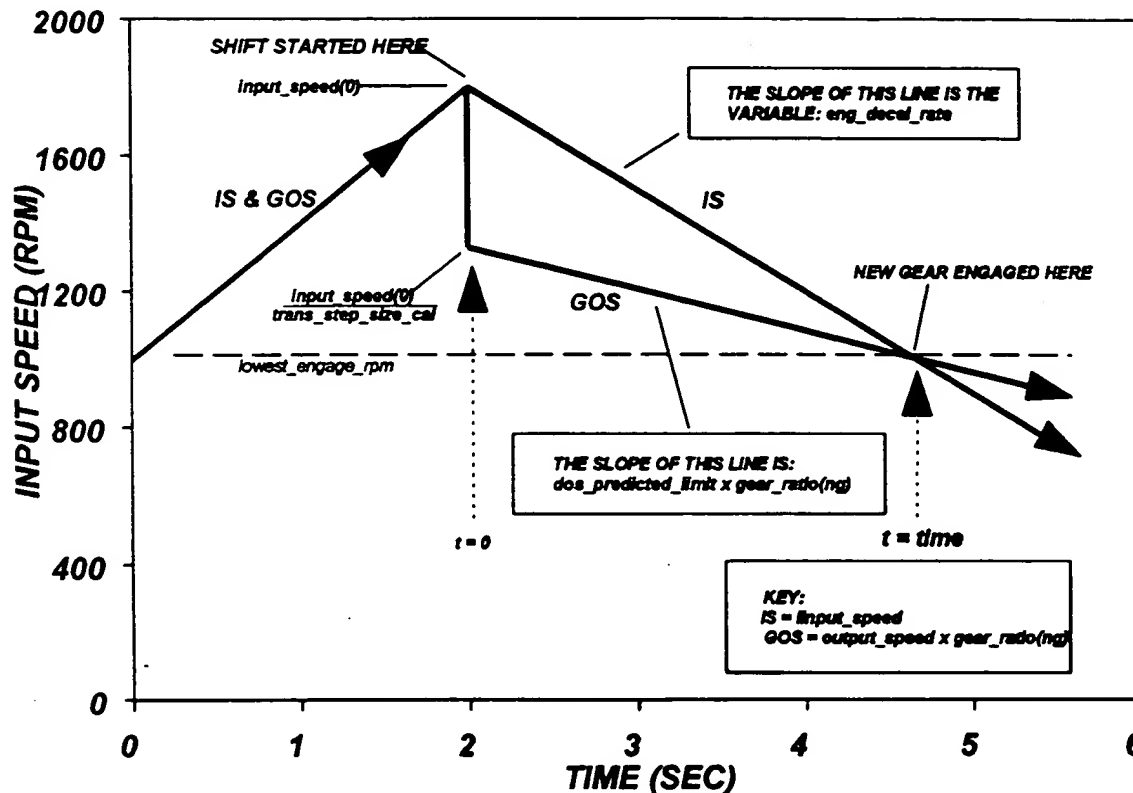Figure 3.2.2.4.2-2.

Figure 3.2.2.4.2-2 - Calculating Dos Predicted Limit

In   Figure   3.2.2.4.2-2,   there   are   two   lines   that   represent
input_speed vs. time and (**output_speed** times *gear_ratio*(tg) of
target gear) vs. time, respectively.   The slope of the **input_speed**
line is known, or learned, by the algorithm.   All other parameters
shown in Figure 3.2.2.4.2-2 are known except dos_predicted_limit.
Writing the equations of the two lines:

lowest_engage_rpm = [(**eng_decel_rate**) · time] + **input_speed**

lowest_engage_rpm= (dos_predicted_limit · *gear_ratio*(tg) · time)
                          + (**input_speed** / **trans_step_size_cal**)

19

Solving the first equation for "time":

time = (*lowest_engage_rpm* - input_speed) / eng_decel_rate

Taking this relation for "time" and plugging it into the second equation, and solving for dos_predicted_limit (rpm/sec):

$$\text{dos\_predicted\_limit} = \frac{\text{lowest\_engage\_rpm} - \dfrac{\text{input\_speed}}{\text{trans\_step\_size\_cal}}}{\text{gear\_ratio(ng)} \times \dfrac{\text{lowest\_engage\_rpm} - \text{input\_speed}}{\text{eng\_decel\_rate}}}$$

where:

input_speed = the transmission input speed at time = 0, before the shift is started.

gear_ratio(ng) = gear ratio of the next gear, from the table of gear ratio calibrations

lowest_engage_rpm = a fixed calibration for the lowest input speed allowable at the completion of the shift. If the algorithm predicts the input speed will fall below this speed, the shift will be inhibited or the engine brake will be used.

eng_decel_rate = this is a variable representative of the expected engine speed deceleration during an upshift. It is "learned" while driving the vehicle (discussed later).

trans_step_size_cal = a variable to be calculated by the software equal to the current ratio divided by the next ratio representing the ratio step size of the next upshift.

In the software, there is actually two "dos_predicted_limits" determined. One limit uses the "learned" natural engine deceleration rate, **eng_decel_rate**, and is called **dos_prdtd_lim_no_jake**. The other limit uses **eng_decel_rate_with_jake**, instead of **eng_decel_rate**. **Eng_decel_rate_with_jake** is a variable determined from an equation in the software that is a function of **torque_accessories**. This

20

linear equation is an approximation and represents the effective deceleration rate of the engine with four (4) cylinders of the engine compression brake activated during an upshift. It is determined empirically. This second limit is called **dos_prdtd_lim_with_jake** in the software.

Note that if a system has variable control of the engine brake (i.e., 2, 4, or 6 cylinders, etc.), more limits can be calculated. Or, if another engine deceleration device is present, such as a transmission-mounted inertia brake, more "engine_decel_rates" must be taken into account and learned. The AutoSplit product currently is planned with only the ability to use an optional engine compression brake (Jake Brake) with the proactive shiftability algorithm.

At this point in the algorithm, the predicted vehicle deceleration - **dos_predicted** - is known along with the two limits. If an upshift is desired, a check is made to see if **dos_predicted** is greater (less negative) than **dos_prdtd_lim_no_jake**. If it is, the upshift is allowed. If it is not, a check is made to see if **dos_predicted** is greater than **dos_prdtd_lim_with_jake**. If it is, the upshift is allowed using the engine brake during that shift. If it is not, the upshift is inhibited until conditions change enough to make the shift feasible.

The issue discussed in the previous section - **vehicle deceleration at zero torque** - is only a concern for splitter-only upshifts in the low gears (i.e., 1-2, 3-4 shifts). The second issue addressed in this section - **required torque at zero acceleration** - is a concern for splitter-upshifts in the higher gears (i.e., 7-8, 9-10 shifts). This addresses problem often referred to as "hunting." The vehicle upshifts, but does not have enough torque in the next gear to accelerate. The vehicle then slows down and downshifts back into the original gear. Figure 3.2.2.4.2-3 illustrates the situation in terms of torque at the wheels.
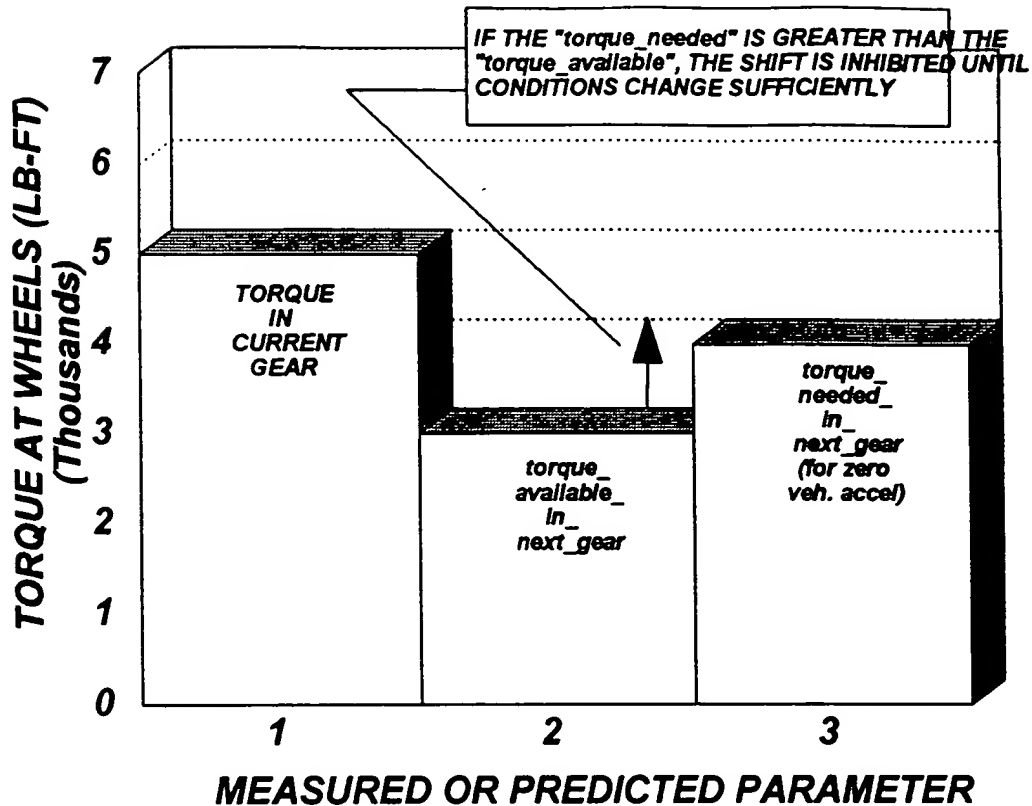
Figure 3.2.2.4.2-3 - Required Torque at Zero Acceleration

The equation used to determine the torque at the wheels required to produce zero acceleration is shown below. Again, it is assumed at this point that the gross combined weight, "W", is known by the algorithm. The procedure to determine "W" will be discussed later in this section.

$$\text{for } A_2 = 0: \qquad T_2 = T_1 - (C \cdot W \cdot A_1)$$

Where: A_1 = vehicle acceleration at time I (ft/sec^2)
T_1 = torque at wheel at time I (lb-ft)
C = wheel rolling radius (ft) / grav. constant (ft/sec^2)
W = gross combined weight (lbs)

At this point, the above equation can be integrated into the vehicle software in a manner similar to that used to determine "A_2" in the previous section. However, to simplify the software and minimize code space, it is recognized that many of the calculations

22

performed in the previous section can be applied to solve the above equation. The appropriate results from the previous section are manipulated and utilized below.

Needed is:

$$T_1 - (C \cdot W \cdot A_1)$$

or,

$$T_2 = T_1 - (C \cdot W \cdot A_1)$$

divide both sides by $C \cdot W$,

$$T_2 / (C \cdot W) = (T_1 / (C \cdot W)) - A_1$$

dividing both sides by $k1\_ability$,

$$\frac{T_2}{C \cdot W \cdot k1\_ability} = \frac{(T_1 / (C \cdot W)) - A_1}{k1\_ability}$$

multiplying both sides by -1,

$$\frac{-T_2}{C \cdot W \cdot k1\_ability} = \frac{A_1 - (T_1 / (C \cdot W))}{k1\_ability}$$

Note that the right side of this equation is exactly **dos_predicted_raw**. Therefore, **dos_predicted** is a filtered version of the quantity needed.

Substituting,

$$\frac{-T_2}{C \cdot W \cdot k1\_ability} = \textbf{dos\_predicted} \ (rpm/sec)$$

Therefore,

$$T_2 = (-1) \cdot \textbf{dos\_predicted} \cdot C \cdot W \cdot k1\_ability$$

In the software, this is stated as:

torque_needed_in_next_gear = (-1) · dos_predicted · *kl_ability* · gcw_cal

| (lb-ft) | (rpm/sec) | (rad-min-ft) (rev-sec) | (lb-sec²) |

where,
    gcw_cal = [W(lb) · roll.rad. (ft)] / grav. const. (ft/sec²)

This represents the torque (T_2) at the wheels needed for zero acceleration (A_2 = 0) in the present vehicle conditions (i.e., present speed, grade, rolling resistance, etc.).

Next, the torque at the wheels available in the proposed next gear needs to be determined. The electronic engine continually reports the engine torque available as a function of engine speed across the J1939 data link. Therefore, if the input/engine speed at the end of the proposed shift can be predicted, the available engine torque at that speed can be interpolated.

To determine the input speed after the proposed shift, the intersection of two lines must be determined. Figure 3.2.2.4.2-4 illustrates this problem. Note that Figure 3.2.2.4.2-4 is similar to Figure 3.2.2.4.2-2.
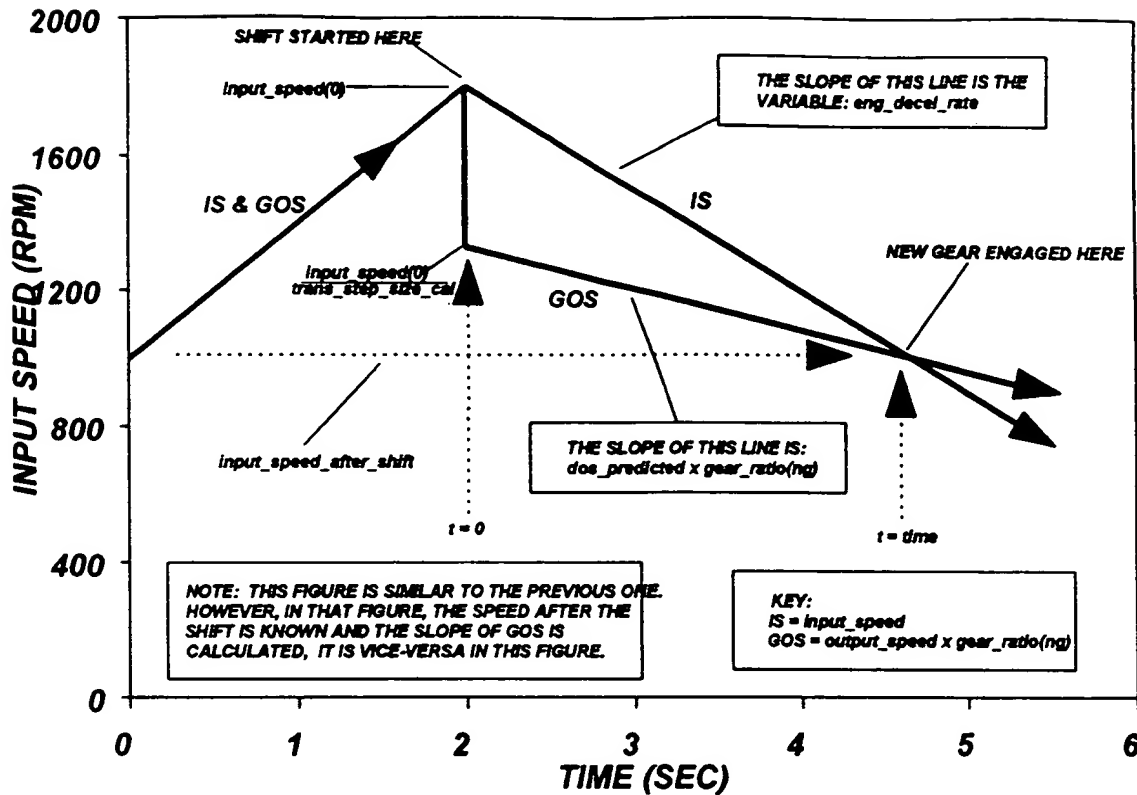
Figure 3.2.2.4.2-4 - Predicting Input Speed After the Shift

The intersection of the two speed curves in Figure 3.2.2.4.2-4 needs to be determined. Again, two linear equations can be written.

$$\text{input\_speed\_after\_shift} = \text{input\_speed} + (\text{eng\_decel\_rate} \cdot \text{time})$$

$$\text{input\_speed\_after\_shift} = \text{input\_speed} / \text{trans\_step\_size\_cal} + (\text{dos\_predicted} \cdot \text{gear\_ratio(ng)} \cdot \text{time})$$

Solving for **input_speed_after_shift**, we derive:

$$\text{input\_speed\_after\_shift} = \text{input\_speed} \times \frac{\dfrac{\text{eng\_decel\_rate}}{\text{trans\_step\_size\_cal}} - [\text{dos\_predicted} \times \text{gear\_ratio(ng)}]}{\text{eng\_decel\_rate} - [\text{dos\_predicted} \times \text{gear\_ratio(ng)}]}$$

Now that the predicted **input_speed** at the end of the shift is known, a software function may be called to determine the available engine torque at that speed. This variable is called the **tang_factor**. "Tang" is an acronym for torque available in next gear. From the software,

tang_factor (%) = pct_trq_avail_at(input_speed_after_shift)
- (nominal_friction_pct_trq - 3)

Note that **tang_factor** is a percentage of the "reference" torque of the engine. Also, "3" is subtracted from the **nominal_friction_pct_trq** (engine friction) to compensate for the fact that **nominal_friction_pct_trq** is not available at a speed different than current engine speed. It was observed empirically that the **nominal_friction_pct_trq** decreases by about three percent over the engine speed decrease that takes place during an upshift. Therefore, subtracting "3" approximates the **nominal_friction_pct_trq** at the **input_speed_after_shift**.

The **torque_available_in_next_gear**, which is the torque available at the wheels, is now calculated. From AutoShift software:

torque_available_in_next_gear (lb-ft) =
[(engine_config.reference_trq · tang_factor)- torque_accessories]
· (gear ratio(ng)) · *axle_ratio_cal ·* **(***margin_cal***)**

where:

engine_config.reference_trq =	a torque value (in lb-ft) reported by the engine upon power-up from which all other torque values reported by the engine (in percent) are referenced.
axle_ratio_cal =	the axle ratio of the vehicle multiplied by .91, which is an approximate overall driveline efficiency

26

margin_cal = a threshold factor to provide a margin of error to the algorithm (note that the algorithm will calculate torque at zero acceleration whereas some positive acceleration after the upshift is actually desired - .65 is a typical value).

Now, at this point, the algorithm compares "torque_needed" to "torque_available" and makes a decision to upshift or hold in gear.

If (**torque_needed_in_next_gear** > **torque_available_in_next_gear**), do not upshift.

As noted twice before in this section, the **gross_combined_weight** (GCW) of the vehicle needs to be known by the algorithm. For applications that have a nearly constant GCW, the value can be a fixed calibration. However, the GCW of most commercial vehicles can vary significantly in service. Note that GCW has been referred to simply as "W" up to this point.

Another form of the shiftability equation can be used to quickly (after a few upshifts) calculate a good approximation (within 2-3,000 lbs.) of GCW. The equation used to calculate GCW is shown here:

$$GCW = W = \frac{(T_1 - T_2)}{[(A_1 - A_2) \cdot C]}$$

where,
C = wheel roll. radius (ft) / grav. const. (ft/sec^2)
W = gross combined weight (lb)
T_1 = torque at wheel at time I (lb-ft)
A_1 = vehicle acceleration at time I (ft/sec^2)

and times 1 and 2 are less than *valid_old_data_time* seconds apart (the **time limit of 4 seconds** was determined empirically during development).

The above equation suggests that if two different wheel torques, T_1 and T_2, and the vehicle accelerations, A_1 and A_2, corresponding to those torques are known, the vehicle weight (GCW) can be calculated. In theory this is true. However, in an actual truck driveline the torsionals, and the heavy signal filtering make any single calculation involving two "points" meaningless.

27

It might then be hypothesized that averaging many calculations over time would "cancel out" the effects of torsionals. This is also true. It took several iterations of calculation methods before a reliable and accurate method was developed. Only this successful method is described here.

Experimentation with truck driveline data shows that the "farther" apart the two torques are in magnitude, the more accurate the calculation becomes. Ultimately, it was found that one accurate and reliable point could be measured when the wheel torque is effectively zero. This, of course, takes place during a shift when the transmission is in neutral. Therefore, if the vehicle acceleration is measured during a shift, then one "point" of the two has been determined: $T_1 = 0$ and A_1 = measured vehicle acceleration.

T_2 and A_2 can then take place when the transmission is back in gear after the shift, engine torque is reapplied, and the vehicle is accelerating. The time between T_1 and T_2 must be less than four seconds. This makes it convenient to record the two points during an upshift.

The method used in the AutoShift software uses this technique. The first point, $T_1 = 0$ and A_1 = vehicle acceleration, is recorded at the instant that the transmission starts to shift into the new gear on upshifts only. Then, T_2 and A_2 can be recorded after the transmission is in gear and sufficient torque has been reapplied. The calculation is done every time step (40 ms) after 20% or more of engine torque is applied for up to four seconds after T_1 and A_1 were recorded. Therefore, GCW is calculated many times and a running average of GCW is also calculated.

This is repeated every upshift and the summation continues to get larger until 1,000 values of GCW have been calculated, summed, and divided by the number of summations. When the summation counter reaches 1,000, the counter is divided by two and the total sum of individual GCW values is divided by two. Thus, the counter initially starts at 0, goes up to 1,000, and then starts back at 500 again.

This "sum halving" process acts as a "running average," or a heavy filter, of the GCW calculations. Note that since no more than about 75 calculations can be done each upshift, after several shifts the counter will be halved and the newly calculated values will have more effect. If a significant change in true GCW takes place, the calculated average will greatly reflect that change within a few upshifts and be quite close to the new true value within a few more upshifts.

There are other methods that could be used to force the average to more quickly reflect a large change in vehicle weight.   For example, if the average of a few of the "new" values differs widely from the large running average, the counter and the total sum can be divided by 4, 8, or more.   This would act like an "adaptive filter."

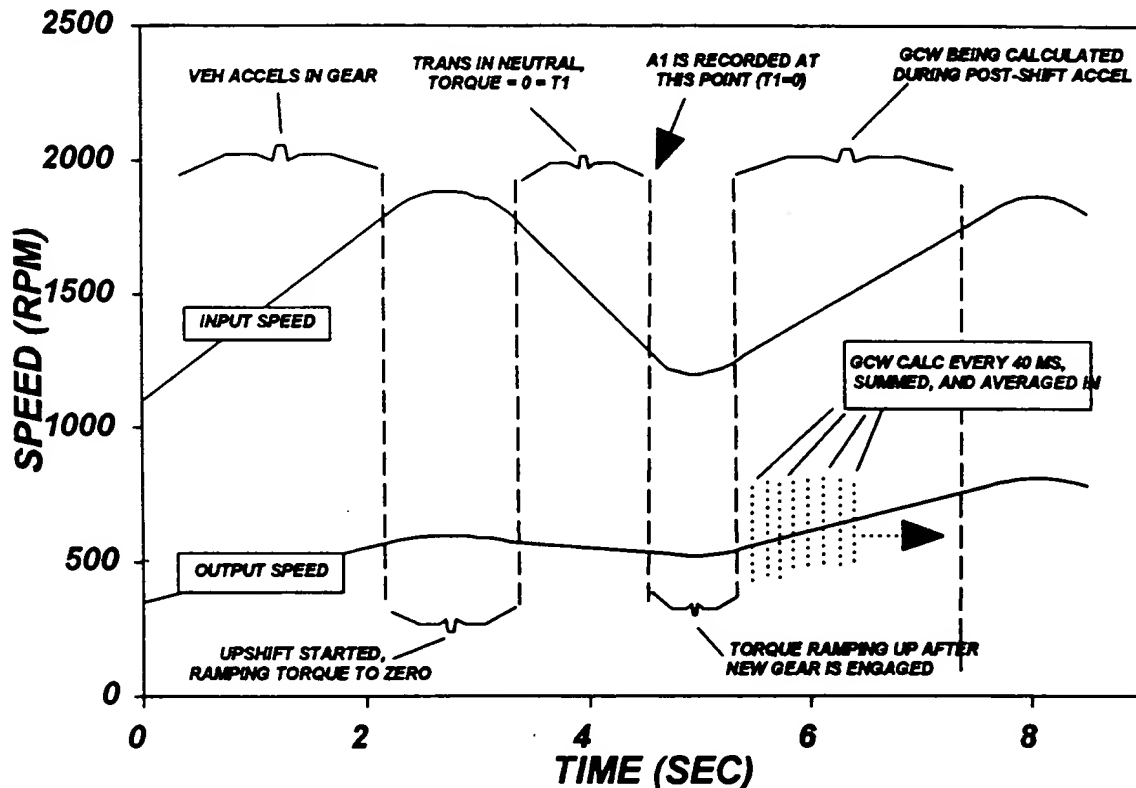The method to calculate GCW is illustrated in Figure 3.2.2.4.2-5.

Figure 3.2.2.4.2-5 - Typical Upshift Where GCW is Calculated

The formula for GCW can be written as:

$$GCW = \sum_{2}^{i} \left\{ \frac{\left[\frac{T_1 - T_i}{(A_1 - A_i) \times C} \right]}{i - 1} \right\}$$

To integrate this routine into vehicle system software, the following equations are used. Again:

$$GCW = W = \frac{(T_1 - T_2)}{[(A_1 - A_2) \cdot C]}$$

However, as calculated above, the filtered version of output shaft acceleration, **dos_filtered**, is known. Remember from above that:

$$A_1 = \textbf{dos\_filtered} \cdot k1\_ability$$

Therefore,

$$GCW = \frac{(T_1 - T_2)}{[(\textbf{dos\_filtered}_1 - \textbf{dos\_filtered}_2) \cdot k1\_ability \cdot C]}$$

If,
$$gcw\_k1 = 1/(k1\_ability \cdot C)$$

and,
$$C = \text{rolling radius (ft)/ grav. constant (ft/sec}^2)$$

then,
$$gcw\_k1 = \frac{\text{grav. const (32.17 ft/sec}^2)}{[k1\_ability \text{ (min-ft/rev-sec)} \cdot \text{roll.rad (ft)}]}$$

where gcw_k1 is a calibrated conversion factor with units of (rev/sec-min-ft). Then:

$$GCW = \frac{(T_1 - T_2)}{[(\textbf{dos\_filtered}_1 - \textbf{dos\_filtered}_2) \cdot gcw\_k1]}$$

Remember that $T_1 = 0$, so:

$$GCW = [-T_2 / (dos\_filtered_1 - dos\_filtered_2)] \cdot gcw\_k1$$

or,

$$GCW = [T_2 / (dos\_filtered_2 - dos\_filtered_1)] \cdot gcw\_k1$$

Therefore, the actual software will calculate:

$$gcw\_local = \frac{torque\_at\_wheels}{dos\_filtered - dos\_filtered\_old} \times gcw\_k1$$

In the software, **dos_filtered_old** is equivalent to **dos_filtered₁** above, where $T_1 = 0$. Note that **dos_filtered_old** is the output shaft acceleration during the upshift at the instant the new gear is engaged. Therefore, a routine in the software monitors **dos_filtered** and records its value at the point the new gear is engaged. Also, an offset called **dos_offset** may be subtracted from **dos_filtered** to get **dos_filtered_old** if the shift is very short in duration. This corrects for filter lag during quick shifts. During the shift, the neutral time is monitored and **dos_offset** gets smaller proportionally and the neutral time gets longer, until **dos_offset** becomes zero.

Gcw_local is the individual GCW calculation during one time step. Each time step, **gcw_local** is summed:

> **gcw_local_total = gcw_local_total + gcw_local**

> **gcw_local_counter = gcw_local_counter + 1**

where **gcw_local_counter** is a counter that keeps track of the number of summations. Then,

> **gross_combined_weight = (gcw_local_total / gcw_local_counter)**

If **gcw_local_counter** > 1000, then, **gcw_local_counter** and **gcw_local_total** are divided by 2.

This algorithm needs to accurately "know" torque at the flywheel. That is, the net torque being transmitted through the input of the clutch or the torque converter needs to be calculated. This "net torque at the flywheel" was previously discussed in this section. Note that accessory torque is a required piece of information. Also, the algorithm needs to "know" the approximate engine deceleration rate during a shift.

31

The gross engine torque and the "base engine" friction torque (engine internal friction and engine-installed accessories, i.e., oil pump) values can be reported from the engine controller, via a communications protocol, such as SAE J1939. However, the "OEM-installed" accessory torque values, such as that from air conditioning, alternator, etc., also need to be known. The engine deceleration rate can be "learned" via an algorithm described later in this section. However, the vehicle needs to be moving and making shifts to get this engine deceleration value.

If a vehicle has a relatively small and/or fairly constant accessory torque loading, it is sufficient to enter the accessory torque value and the engine deceleration rate as a constant. However, if the accessory torque loading varies widely and is significant, it must be measured as it changes. Also, if the engine deceleration rate varies significantly, it must be measured as it changes. It has been determined that accessory torque and engine deceleration rate vary dependently on each other. If the accessory load increases, the engine deceleration rate increases (gets more negative) in proportion to it, and vice-versa.

This portion of the algorithm provides a method to measure and/or calculate accessory torque and engine deceleration rate with the vehicle stopped or moving.

There are two "modes" of this algorithm: 1) Vehicle stopped with engine idling and transmission in neutral, and 2) Vehicle moving. These two modes are separate, but complementary. Each is described below.

1) Vehicle Stopped, Engine Idling, Trans in Neutral
During this case, the accessory torque is identical to the net torque being reported by the engine (net torque = gross torque minus engine friction torque. This value is recorded and "filtered" each time step. Then, the **engine_decel_rate** is determined from this accessory torque by a linear equation in which **engine_decel_rate** is a function of accessory torque. This equation must be inputted into the software. This relation is illustrated below in **Figure 3.2.2.4.2-6.**

NOTE: THE "engine_decel_rate" VALUE
DETERMINED HERE IS AN APPROXIMATION
FOR THE SYSTEM TO GET STARTED WITH.
"engine_decel_rate" IS MEASURED AND UPDATED
AS THE VEHICLE DRIVES AND MAKES UPSHIFTS.

engine_decel_rate = -385 + (-2 x torque_accessories)
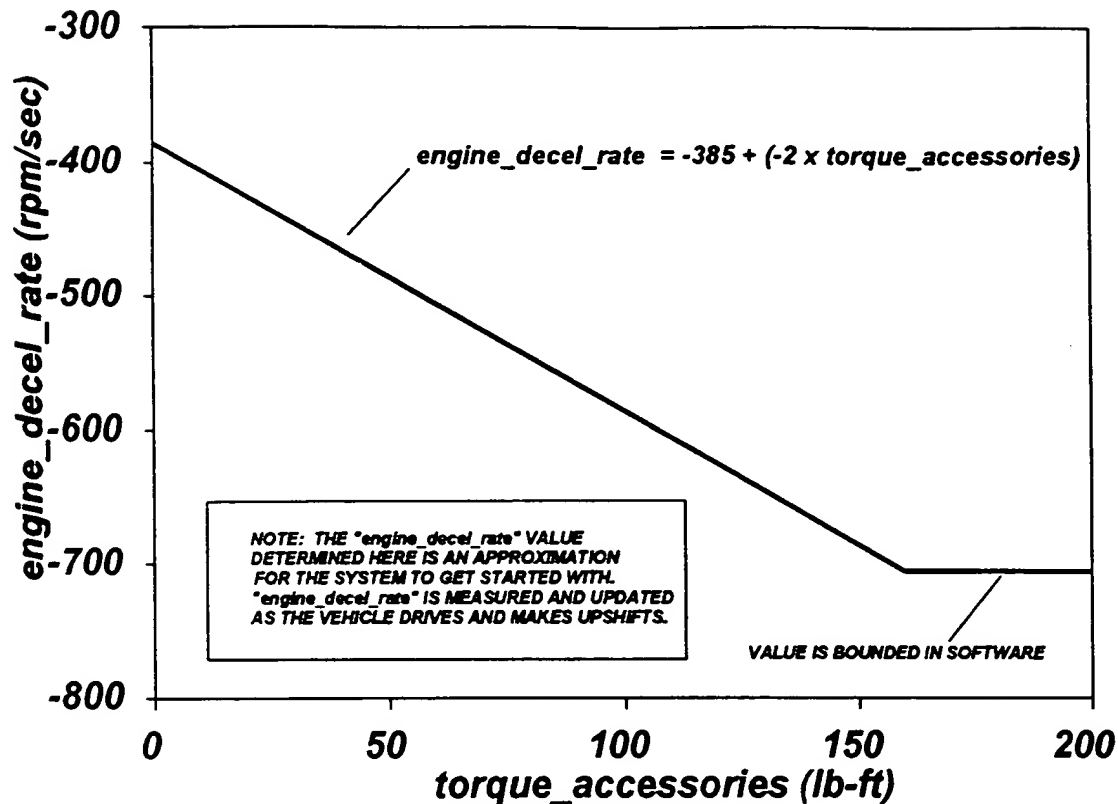
VALUE IS BOUNDED IN SOFTWARE

Figure 3.2.2.4.2-6 - Engine Decel Rate vs. Accessory Torque

The method for "vehicle stopped, idling, in neutral" can be
summarized as listed here:

If the vehicle is stopped, the throttle position is zero, the
transmission is in neutral, and the engine speed is less than
1100 rpm (high idle), then:

torque_accessories (lb-ft) = net_engine_pct_trq

engine_decel_rate (rpm/sec) = A + (B · **torque_accessories**)
                             = -385 + (-2 · **torque_accessories**)

where A and B are "calibrations" programmed into the software
for the vehicle-engine configuration, and -385 and -2 are
calibrations for the Detroit Diesel Series 60.

33

It should be noted that the **engine_decel_rate** determined above is an approximation for the system "to get started with" and is corrected and updated as soon as the vehicle gets moving and making upshifts.

2) Vehicle Moving

Once the vehicle is moving, the **engine_decel_rate** is measured and updated during upshifts. A description of this **engine_decel_rate** measurement method is provided later in this section. Basically, the software measures the transmission input speed decel rate during an upshift while the transmission is in neutral during the upshift.

Then, **torque_accessories** is calculated from the same linear equation as used in mode one above. This linear equation is inverted, of course, to make **torque_accessories** a linear function of **engine_decel_rate**. This method is summarized below.

> If the shift type is an upshift, and no "engine decel aids" are being used for the upshift (i.e., Jake Brake, engine inertia brake, etc.), then the **engine_decel_rate** is determined, and filtered, and:

$$\textbf{torque\_accessories} = C + (D \cdot \textbf{engine\_decel\_rate})$$
$$= -192.5 + (-.5 \cdot \textbf{engine\_decel\_rate})$$

> where C and D are "calibrations" programmed into the software and are related to the A and B calibrations, since the same linear equation used in mode one above is used, but the equation is modified to reverse the dependent and independent variables. Note that 385 (A) divided by 2 is 192.5 © and .5 (D) is the reciprocal of 2 (B).

Another option would be to have the transmission system "learn" this equation. This is possible, but still needs to be developed. It would reduce the vehicle-configuration dependency of the algorithm by making it more adaptive. In other words, it is possible for the software to learn the linear proportional relation between **torque_accessories** and **engine_decel_rate**.

To measure **engine_decel_rate**, the software monitors **input_speed** only during upshifts that do not use the engine brake. The software records the **input_spd_first_point** when neutral is obtained and starts a timer. Then, when the new gear is engaged, the "second speed point" is observed and the **engine_decel_rate** for that shift is now calculated. A light filter is used on the result that takes one-eighth of the "latest" decel rate and adds it to

34

seven-eighths of the running average. Note that the average has a good initial value since **engine_decel_rate** is estimated when idling in neutral with a linear equation that is a function of **torque_accessories**.

3.2.2.5 Anti-Hunt Strategy

After a shift is completed, the system will start a timer and offset the next shift point to prevent hunting after splitter-only shifts based on the following conditions:

While the timer is less than *offset_time* and the last shift was a downshift, then the **upshift_point** becomes (**auto_up_rpm** + *up_timer_offset_rpm*). While the timer is less than *offset_time* and the last shift was an upshift, the **downshift_point** becomes (**auto_dn_rpm** - *dwn_timer_offset_rpm*).

When the timer exceeds *offset_time* and an upshift has just been completed, then the new **downshift_point** becomes (**auto_dn_rpm** - *dwn_offset_rpm*). The **downshift_point** reverts back to **auto_dn_rpm** when **input_speed_filtered** exceeds (**auto_dn_rpm** + *dwn_reset_rpm*).

When the timer exceeds *offset_time*, and a downshift has just been completed, then the new **upshift_point** becomes (**auto_up_rpm** + *up_offset_rpm*). The **upshift_point** reverts back to **auto_up_rpm** when **input_speed_filtered** falls below (**auto_up_rpm** - *up_reset_rpm*).

A similar strategy is used during lever shifts to move the opposite "ok-to-shift" point away during a lever shift. For example, a lever upshift is started and the transmission is in neutral. The lever downshift point is moved away from the upshift point in the same manner as for splitter shifts above. This will help prevent the system from changing the target gear during a lever shift if the vehicle decelerates somewhat.

3.2.3 Shift Process

This, section repeats the shift sequence information of section 3.0.2 and then describes in more detail the system functions during that process. Normal AutoSplit operation consists of both driver-initiated (lever) shifts and system-initiated, automatic (splitter) shifts. The normal shift sequence for each of these shifts is described below. Every shift process is divided into three phases: Predip, Sync, and Recovery.

Splitter-Only Shift
>Predip phase begins here<
1) The system detects the optimal time to shift based on load, **input_speed**, etc. At this instant, the system overrides

cruise control, engine brakes, throttle, etc. via J1939 engine override commands, commands the splitter to neutral, and modulates the engine torque to allow splitter disengagement via J1939.

>Sync phase begins here<

2) The system confirms splitter disengagement via a comparison of the ratio of **input_speed** to **output_speed** with the transmission ratio table. At this instant, the system implements anti-hunting routines, and modulates **engine_speed** via J1939 to synchronize the splitter for the target ratio.

3) When the system senses impending synchronous via **input_speed** and **output_speed** signals, it commands the splitter towards target gear engagement.

>Recovery phase begins here<

4) The system confirms splitter engagement via a comparison of the ratio of **input_speed** to **output_speed** with the transmission ratio table. At this instant, the system commands the engine to reapply torque via J1939.

5) When torque has been reapplied, the engine resumes control of the throttle, engine brakes, cruise control, etc.

Lever/Splitter Shift

1) The driver display flashes the available lever position to indicate it is "OK" to shift the lever to that position.

>Predip phase begins here<

2) When a lever shift is desired, the driver pulls the lever to neutral while activating the intent-to-shift feature (TBD – either a momentary button, or a force detent in the knob or lever). At this instant, the system overrides the cruise control, engine brakes, throttle, etc. via J1939, commands the splitter to neutral, and modulates the engine torque to allow splitter and lever disengagement via J1939.

>Sync phase begins here<

3) The system confirms transmission neutral via a comparison of the ratio of **input_speed** to **output_speed** with the transmission ratio table, and by the lever position sensor. At this instant, the system implements anti-hunt routines, commands the splitter to the engaged position for the new ratio, and modulates the **engine_speed** via J1939 to synchronize the transmission for the target gear ratio.

4) The driver moves the lever into the new position.

>Recovery phase begins here<

5) The system confirms the new gear engagement via a comparison of the ratio of **input_speed** to **output_speed** with the transmission ratio table. At this instant, the system commands the engine to reapply torque via J1939.

6) When torque has been reapplied, the engine resumes control of the throttle, engine brakes, cruise control, etc.

3.2.3.1 Predip Phase

Upon entering the Predip phase of the shift, the AutoSplit system will first temporarily disable the engine brakes then override throttle and cruise control, by entering the Engine Torque Control mode of J1939. For <u>splitter-only shifts</u> this occurs as soon as the system determines it is time for an automatic splitter shift. For <u>lever shifts</u> this occurs when the intent-to-shift switch is activated (<u>if</u> it is activated).

The system will then command the splitter to change state and enter the Engine Torque Control mode and perform torque modulation in accordance with Figure 3.2.3.1-1. Then, if neutral is still not detected, the torque "pulses" begin – starting with a "zero driveline torque" pulse. The gross engine torque needed to create zero drive line torque is a dynamic variable periodically calculated by the system, and is a function of gross engine torque and engine friction (both reported by the engine via J1939), accessory torque (continuously calculated by the system), and engine acceleration. Note that if the gross engine torque is less than *open_throttle_pct* (typically 5 percent), the Predip phase will commence directly with the "zero drive line torque" pulse as shown in Figure 3.2.3.1-1 (i.e., coasting downshifts).

The algorithm needed to calculate the gross engine torque needed to create zero driveline torque is as follows.

$$\text{needed\_percent\_for\_zero\_flywheel\_trq} =$$
$$\text{torque\_accessories} + \text{nominal\_friction\_pct\_trq}$$
$$+ \text{torq\_to\_accel\_eng}$$

The Predip phase **torque_ramp_off_rate** will be adjusted as a function of gross combination weight (GCW) using the GCW value determined in the shiftability calculations. A typical value of **torque_ramp_off_rate** is 1% per 10 msec. This will provide smooth operation across the range of GCW's. A default GCW representative of the average of the possible GCW's (50,000 lbs.) will be used upon power-up and until a sufficient approximation of GCW can be calculated (after a few upshifts).
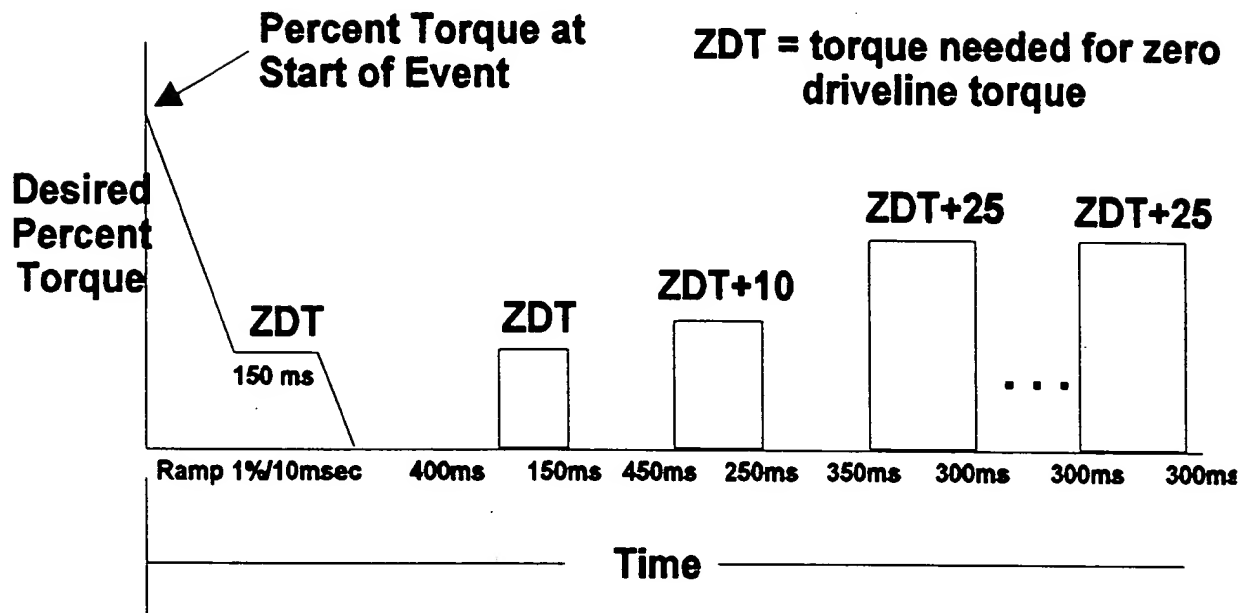
Figure 3.2.3.1-1 - Predip Phase Torque Control

Transmission disengagement will be confirmed when the software indicates that the current gear ratio has disengaged using the **input_speed** and **output_speed** variables. For splitter-only shifts, this disengagement is due to the splitter moving to neutral. For lever shifts, it is either the splitter, or the lever, or both, that causes the system to sense disengagement. If the intent-to-shift switch is activated, the splitter almost always comes to neutral before the lever is brought to neutral. However, if the driver pulls the lever to neutral without activating the intent-to-shift switch, the Predip phase doesn't start until neutral is sensed, at which point the system goes immediately to the Sync phase.

Again, the Predip phase is complete as soon as the software detects transmission neutral. Note that during lever shifts, the shift

lever may be moved to neutral during either the Predip or the Sync phase. Engagement and disengagement are sensed by using the following equation:

$$trans\_sync\_error = input\_speed\_filtered - (output\_speed\_filtered \cdot gear\_ratio)$$

The software is continually calculating the **trans_sync_error** for each gear ratio and then checking to see if the absolute value of this variable is less than *trans_window_calc* (typically 30 rpm) for that specific ratio. To declare neutral (when in gear), the software must see the absolute value of **trans_sync_error** larger than *trans_window_calc* for a time of *gear_out_time*. Similarly, to declare an engaged gear (when in neutral), the software must see the absolute value of **trans_sync_error** smaller than *trans_window_calc* for a time of *gear_in_time_auto* for splitter-only shifts or *gear_in_time_lever* for lever shifts .

3.2.3.2 Sync Phase

After confirming transmission disengagement, the shift will enter the Sync phase of the shift. The AutoSplit system will command Engine Speed Control mode via J1939 and request the engine to go to the calculated synchronous speed of the target gear based on current **output_speed**. Also upon entering the Sync phase, for <u>lever shifts</u> the splitter is commanded to engage the new ratio as soon as lever neutral is sensed. For <u>splitter-only shifts</u>, the splitter is commanded to engage the new ratio when synchronous for the new gear is sensed. Note that reengaging the splitter in the Sync phase as soon as lever neutral is seen prevents the driver from engaging the wrong new lever position, since the front box "grinding" will occur.

Note that when the engine is commanded to synchronous speed for the new gear, it is commanded to a speed that is a fixed amount "away" (*sync_offset*) from synchronous for that gear. Note that if nearly exact synchronous speed was commanded, the engagement detection software function would "think" that the transmission is in a gear. Therefore, the engine is commanded to a speed either "above" or "below" (depending upon shift conditions) synchronous that would achieve *sync_offset* rpm (typically 35 rpm) of speed difference across the intended engaging dog clutch and gear. Then, the engagement detection software function would be "looking" for a synchronous speed between + or - *trans_window_calc* (see above) across the engaging dog clutch to sense engagement.

The Sync phase is complete when the system confirms engagement of the new gear using the engagement detection software function as described above. If engagement is not confirmed by *sync_time*, the

system will command opposite splitter state and repeat the synchronizing process for a maximum of *split_attempts* before setting an "Unable to Select Splitter Direct/Indirect" fault and entering a degraded mode of operation.

3.2.3.3 Recovery Phase

Once engagement of the new gear is confirmed, the engine control is gradually returned to the engine/driver as follows. If **pct_demand_at_cur_sp** is greater than *open_throttle_pct* (typically 5 percent), the Engine Torque Control mode torque limit is set to **needed_percent_for_zero_flywheel_trq**, then ramped up from there at a rate of **torque_ramp_up_rate** for that gear ratio, which also will bee adjusted as a function of GCW for smooth operation at all possible GCW values. A typical value of **torque_ramp_up_rate** is 1% per 10 msec. Again, if **pct_demand_at_cur_sp** is less than *open_throttle_pct* percent, the Engine Speed Control mode is invoked and the desired speed is set to the current engine speed, then ramped down in speed from there at a rate of *speed_ramp_off_rate*. This speed ramping on downshifts prevents the sharp negative drive line pulse due to engine motoring.

The Recovery phase **torque_ramp_up_rate** will be adjusted as a function of vehicle weight (GCW) using the GCW value determined in the shiftability calculations. This will provide smooth operation across the range of vehicle weights.

After recovery is complete, J1939 engine override control is returned to Engine Follower mode (no override).

3.2.3.5 Speed Calculations

Input_speed and **output_speed** calculations are used continuously throughout the AutoSplit algorithms. **Input_speed** is derived from the **engine_speed** value as reported from the engine via J1939. **Output_speed** is derived from the transmission-mounted output speed sensor and will be calculated on a 25ms maximum time interval. The most time critical calculations occur during the synchronizing process. A single pole filter is required for the **output_speed** calculation. Further filtering of these signals is done as necessary for specific software functions that require heavier filtering. Fault detection and handling is covered in sections 8.3 Fault Detection and Fault Tolerance.

3.2.4 Special Cases

Any scenarios that are not a fault condition but require special action, are discussed in the following sections.

3.2.4.1 Tire Skid

Whenever a filtered version of **output_speed_accel** (filter amount to be determined) is less than (-)*skid_limit*, then:

1. Save the last valid **output_speed** prior to the skid condition.
2. If the determine_gear software function indicates a gear is engaged, stay in that gear.
3. If the function indicates neutral, stay in neutral.

Continue with steps 1 - three until (**output_speed**/*skid_timer_const*) time has expired, or **output_speed_accel** is greater than (*skid_limit*/2), or **output_speed** ≤ *min_output_spd* and stable for five seconds.

3.2.4.2 Unexpected Neutral

Since a driver is an integral "input" to the control system, situations may arise (i.e., after a lever shift, etc.) in which the lever may be reengaged but the transmission is neutralized due to the splitter grinding, range still on the block, etc. To ensure reengagement under these conditions, every *re_sync_time* (2 sec typical) seconds while in the Sync phase, the engine is commanded to momentarily pass through synchronous speed and then back to the normal *sync_offset* rpm below synchronous. Confirmation of gear engagement is suppressed during this time. This will "unblock" the range and/or the splitter.

3.2.4.3 Wrong Lever Position Engagement

If the driver attempts to engage a different lever position than the system is synchronizing for (and the display is flashing), usually the transmission will just grind and block the lever from engaging the gear. However, if the driver depresses the clutch far enough to engage the clutch brake, the lever generally can be engaged in the wrong position. To prevent the system from going into the Predip phase indefinitely, the system will check for this condition (current gear not equal to destination gear) every time it confirms engagement of a new gear.

To sense this, when in neutral the gear engagement sensing routine looks for engagement (for one time step) of a ratio other than the **destination_gear_selected**. If this condition exists during that neutral period for *jammed_time* (typically 2 seconds), the software declares the engaged gear the new gear and places the system in Recovery phase.

3.2.4.4 Long Periods of Neutral During Vehicle Movement

If the driver leaves the lever in neutral for a long period of time (longer than the normal 1 to 2 seconds), the shift process will remain in the Sync mode until gear engagement is confirmed. During

this time, the system will continue to command the engine to the synchronous speed minus *sync_offset* indefinitely if the driver has the throttle depressed. If the throttle is not depressed, the system will switch to Engine Follower mode (no override) after *maintain_sync_time* seconds (typically 3 seconds) in the Sync phase. The system will resume active engine control - resuming the Sync phase and commanding the engine to synchronous speed minus *sync_offset* for the target gear flashing on the display when the throttle is again depressed.

4 COST, RELIABILITY AND DUTY CYCLE
Refer to the AutoSplit Product Design Specification.

5 ENVIRONMENT
Refer to the AutoSplit Product Design Specification.

6 INTERFACE REQUIREMENTS
Refer to the AutoSplit Product Design Specification.

7 MANUFACTURING, PURCHASING, AND QUALITY

7.1 Final Assembly and Calibration

7.1.1 Final Subassembly Programming
AutoSplit calibration and configuration parameters will be programmed into the ECU at the time of the final subsystem assemblies. A default set of configuration parameters particular to the vehicle and engine type will be programmed into the ECU at the end of the ECU final assembly. These parameters consist of shift point selection parameters and vehicle configuration parameters.

Transmission specific configuration parameters will be programmed into the ECU at the end of transmission final assembly. These parameters will consist of front box and back box gear ratios, back box range and splitter positions, and speed sensor requirements (number of teeth per revolution). Transmission specific calibration functions will be performed and the resulting calibration parameters will be loaded into the ECU prior to the end-of-line transmission test.

7.1.2 AutoSplit System Final Assembly Programming
Final calibration and configuration parameters particular to the

vehicle and engine type will be programmed at the OEM end-of-line checkout once vehicle specifics are known. These parameters will be programmed according to SAE J2214 recommended practice titled "OEM/Vendor Interface Specification for Vehicle Electronic Programming Stations.

7.2 Final Test and Certification
Refer to the AutoSplit Product Design Specification.

8 SERVICE

8.1 Service Level
Refer to the AutoSplit Product Design Specification.

8.2 Serviceability and Service Time

8.2.1 On-Board Diagnostics
On-board diagnostics will be performed on a continuous basis during operation. The diagnostic routines will check all sensors, and solenoids. See section 8.3, Fault Detection and Fault Tolerance for additional information.

8.2.2 Service Time
Refer to the AutoSplit Product Design Specification.

8.2.3 Diagnostic Tests
None.

8.3 Fault Detection and Fault Tolerance
Faults are set and specific action is taken to keep the vehicle operational, taking into account the ability to continue operation without further damaging the transmission or inducing additional mission disabling conditions. Degraded mode response to the faults indicated below are found within the fault description below. All active faults that indicate a component failure turn on solid or flash the "Service Transmission Lamp." If a fault occurs during a shift, unique action required to complete the shift is specified below as well.

8.3.1 System Battery Voltage Fault
This fault is set whenever battery voltage is detected to be high (Vbatt > 18v), weak (9 < Vbatt < 11), or low (Vbatt < 9). The system goes to "hold in gear with no engine override" until the Vbatt is returned to normal. The transmission may be driven as a manual 5-speed.

8.3.2 Ignition Voltage Fault

This fault is set whenever ignition voltage is detected high (Vbatt > 18v). Result same as 8.3.1.

8.3.3 ECU Fault
This fault is set whenever the system ECU calculates an incorrect RAM, ROM, or EEPROM checksum. The system holds in the present gear with no engine override until the problem is corrected (i.e., after a power-down and power-up). The transmission may be driven as a manual 5-speed.

8.3.4 Output Shaft Speed Sensor Fault
This fault is set whenever the system detects an open or short in the speed sensor circuit. The system holds in the present gear with no engine override until the problem is corrected. The transmission may be driven as a manual 5-speed.

8.3.5 Unable to Select Splitter Indirect Fault
This fault is set whenever the splitter has failed to complete to indirect after *split_attempts* splitter attempts. The system holds in the present gear with no engine override until the problem is corrected. The transmission may be driven as a manual 5-speed.

8.3.6 Unable to Select Splitter Direct Fault
This fault is set whenever the splitter has failed to complete to direct after *split_attempts* splitter attempts. The system holds in the present gear with no engine override until the problem is corrected. The transmission may be driven as a manual 5-speed.

8.3.7 Ignition Switch Turned Off Fault
This fault is set whenever ignition voltage is not detected during vehicle movement. The system holds in the present gear with no engine override until the problem is corrected. The transmission may be driven as a manual 5-speed.

APPENDIX A - CALIBRATION PARAMETERS

Below is a draft list of calibration and/or EEPROM parameters
required for AutoSplit control. An effort should be made to reduce
or "hard code" as many as practical.

This section is incomplete.